Multivariable Function Interpolation using a GMDH Network

Michael C. King and Gary L. Lebby
Machine Intelligence and Power Associated Research Group
Department of Electrical Engineering
North Carolina Agricultural and Technical State University
Greensboro, NC 27411

Abstract

A Group Method of Data Handling (GMDH) Network has been designed to perform multidimensional approximation by using multiple networks. The network is designed to interpolate a given multivariable function using data sets reduced in size. The ability of the GMDH network to fit the error of other subnetworks without overfitting its own error is examined. This paper will demonstrate the GMDH network's ability to approximate various single-valued multivariable functions using multiple networks.

1 Introduction

The GMDH network is a multinomial network that was introduced in 1968 by A. G. Ivakhnenko. Ivakhnenko developed the GMDH network to provide a more effective means for predicting the number of fish in rivers and oceans. The Network worked well for its original purpose and it also worked well with many other endeavors. But, the GMDH network was not widely known until Ivakhnenko's paper in 1971 [1].

The original GMDH method has a design deficiency due to the fact that the network coefficients are approximated in isolation[2] In order to effectively train a GMDH network, a large data set containing multiple events is needed. Therefore this method of training could use a large amount of memory. Maybe the amount of data needed could be beyond the range available. In an attempt to reduce the amount of data needed to successfully train a GMDH network, the method of creating sub-nets to train a final GMDH network to accurately approximate a multivariable function is examined.

2 The GMDH Network Architecture

The group method of data handling (GMDH) uses processing elements (Figure 2) that transform two inputs \( x_1 \) and \( x_2 \) into an output \( y \) by using a second-degree multinomial as follows [1]:

\[
y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1 x_2
\]

GMDH networks are for the most part recurrent or feedforward (Figure 2) such that the components are connected in a series-parallel fashion. GMDH networks are trained in a two stage heuristic process [3]. The first stage is one of self organization and the second phase is the adaptive phase. The networks are defined by [4]: (1) the number of sub-networks; (2) the number of layers per network; (3) the number of processing elements per layer and (4) the connectivity between the outputs of a preceding layer and the current layer. All the above items with the exception of the connectivity are chosen at the start of the problem by the end-user. The connectivity of the network is determined during the self organization phase.
where \( n \) is the number of training events. The solution of the above system may be expressed in matrix form as:

\[
A = (X^T X)^{-1} X^T Y
\]

where the matrix \( A^T = (a_0, a_1, a_2, a_3) \); the matrix \( Y^T = (y_1, y_2, \ldots, y_n) \) and the matrix

\[
X = \begin{pmatrix}
1 & x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\
1 & x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nn}
\end{pmatrix}
\]

for the \( l \)-th and \( m \)-th input pair under consideration. All unique pairs of inputs are processed and the sum square error is calculated for each input pair. The top \( L_1 \) input pairs (those pairs whose model output has the least sum square error in all data sets) have their outputs passed on to the next level. This process ends if succeeding levels do not further reduce the sum square error or the user-defined number of layers for the network has been reached.

**Figure 2:** GMDH Multiple Network Configuration

### 2.1 GMDH Self Organization Phase

The concepts of self-organization imply an ability to configure a given internal structure influenced by external conditions. Given \( n \) independent variables \( x_1, x_2, \ldots, x_n \) and one dependent variable, \( y \), the self organization phase proceeds as follows:

1. Divide the data into three sets, one for training the network and the other two to serve as independent testing sets to monitor overfitting.

2. From the set of \( n \) independent variables,

\[
L_{\text{max}} = \left( \frac{n}{2} \right)
\]

a maximum of \( L_{\text{max}} \) combinations are possible. The maximum number of processing elements in the first layer (\( L_1 \)) is less than \( L_{\text{max}} \). There are four unknown coefficients \( (a_0, a_1, a_2, a_3) \) that are obtained from the following equations:

\[
y_1 = a_0 + a_1 x_{11} + a_2 x_{12} + a_3 x_{11} x_{12} + a_4 x_{13} x_{14}
\]

\[
y_2 = a_0 + a_1 x_{21} + a_2 x_{22} + a_3 x_{21} x_{22} + a_4 x_{23} x_{24}
\]

\[
y_3 = a_0 + a_1 x_{31} + a_2 x_{32} + a_3 x_{31} x_{32} + a_4 x_{33} x_{34}
\]

\[
y_n = a_0 + a_1 x_{n1} + a_2 x_{n2} + a_3 x_{n1} x_{n2} + a_4 x_{n3} x_{n4}
\]

### 3 GMDH Multinetwork Investigation

The Multinetwork configuration is used here in an attempt to reduce the amount of events needed in the data set to train a GMDH network. It is also used to fit the error of the three other networks while monitoring the danger of overfitting. A restricted GMDH network, written by G. Lebby will be used to restrict the number of layers that compose the network [4].

#### 3.1 Restricted GMDH Network

In a restricted network the number of layers is equal to the number of user-defined inputs. The first layer will also contain as many processing elements as there are inputs. The number of inputs passed on to the next layer will decrease by one until only two inputs and a single processing element is left. This in turn will force the network to look like a cone. To increase the accuracy of the network, the number of inputs will have to be increased or the network will have to be trained using a very large data set.

The restricted GMDH network is used because in conventional GMDH networks, layers are added until the mean squared error is begins to increase. Therefore, the effects of multiple nets using a conventional
network may not be easily seen. The restricted GMDH also reduces the size of the network to a size that is predetermined by the user.

3.2 Data set of Events

The size of the data sets used to train the networks will be reduced from a desired one thousand events down to one hundred events. The function will consist of three independent variables, \( x_1, x_2, \) and \( x_3 \). These variables will be generated at random. Because the network is a restricted GMDH network, it is known that three variables will not be enough to properly train the network. Therefore, three more variables will be artificially replicated to give a total of six independent variables \( \{x_1, x_2, \ldots, x_6\} \). The three other variables were replicated by squaring the first three variables. The first three variables will be entered into the following equation:

\[
f(x_1, x_2, x_3) = x_1 + 4x_1x_2 + x_3^2 - 3x_1x_3 \tag{6}
\]
to produce to desired output \( y \). The data set of events consists of seven variables and one hundred events as follows:

\[
\begin{align*}
x_1, x_2, & \ldots, x_6, y_1 \\
\vdots \\
x_{1n}, x_{2n}, & \ldots, x_{6n}, y_n
\end{align*}
\tag{7}
\]

where \( n \) is the number of events in the data set. This procedure will be repeated four times resulting in four data sets.

3.3 GMDH Multi-net Architecture

The GMDH multiple network configuration will differ from some of the more common multi-net configurations in such a way that it will not sum the outputs from the subnets to produce an approximation for the output \( y \). This type of architecture uses the approximations of the subnets to produce a training set for the final GMDH network. The final GMDH network is expected to fit the error of the sub-networks. As stated before, there is also an increased possibility of over-fitting the network. The multiple network configuration presented here will consist of a total of four networks (three sub-networks and the final error fitting network) with each of the sub-networks being trained using different data sets. The fourth data set will be used to test the three sub-networks passing there results to a data file containing the actual results. This data set will be used to train the final net.

4 Investigation

The first data set, \( dat_1 \) will be loaded to produce \( NET_1 \). Then \( dat_2 \) and \( dat_3 \) will be loaded to produce, \( NET_2 \) and \( NET_3 \), respectively. Load \( dat_4 \) and run it through the \( NET_1, NET_2, \) and \( NET_3 \). The results produced by the three sub-nets will be stored in a final data file (FIN.DAT). This data will be used to train FINALNET. The fact that there were six independent variables used to train the sub-nets, means that each sub-network will possess a total of six layers and twenty one processing elements. FINALNET will only contain three layers and six processing elements. No variables were replicated artificially in \( FIN.DAT \) because the outputs produced by the sub-nets should be close enough to the actual value such that the final network (FINALNET) should be able to self-organize given a set with one hundred events.

5 Concluding Remarks

The method of creating multiple GMDH networks has room for improvement. It only proved to be successful in its ability to fit the error for some of the sub-networks values. This may be a result of the lack of layers the network was able to use. This method will be examined again using more replicated variables with the same amount of events included in the data set.

References

